UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/637,078	08/11/2000	Erik R Altman	YOR9-2000-0415US1 (8728-4	8733

46069 7590 10/06/2005

F. CHAU & ASSOCIATES, LLC
130 WOODBURY ROAD
WOODBURY, NY 11797

EXAMINER
WOOD, WILLIAM H

ART UNIT	PAPER NUMBER
2193	

DATE MAILED: 10/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

10

Office Action Summary	Application No.	Applicant(s)	
	09/637,078	ALTMAN ET AL.	
	Examiner	Art Unit	
	William H. Wood	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 July 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-16,18-30 and 32-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-16,18-30 and 32-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claims 1, 3-16, 18-30 and 32-42 are pending and have been examined.

Response to Amendment

The finality of that action is withdrawn and prosecution is reopened.

Claim Rejections - 35 USC § 102/103

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are

such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made

3. Claims 1, 4-8, 11-13, 16-17, 22, 38 are rejected under 35 U.S.C. 102(e) as anticipated by **Krishnaswamy** et al. (USPN 6,622,300) or, in the alternative, under 35 U.S.C. 103(a) as obvious over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354).

Claim 1

Krishnaswamy's background disclosed a method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy *(column 1, lines 10-43)*, comprising the steps of:

- executing a computer program *(column 1, lines 33-35)*;
- storing, in a memory array, profile counts for a plurality of events associated with the execution of the computer program *(column 1, lines 36-39)*
- selecting at least one of the plurality of events for profiling *(column 6, lines 24-28, at some point in time the events to be profiled are selected)*;
- updating the profile counts for only the events *(column 1, lines 33-37)*; and
- assisting compilation and optimization of the computer program, based upon the profile counts stored in the memory array *(column 1, lines 33-43)*.

Depending on interpretation of "the memory array being separate and distinct from the memory hierarchy", **Krishnaswamy** may have anticipated the claimed invention through column 5, lines 35-45, column 6, lines 34-36 and figure 2, elements 70, 100,

110, 160 and 170. The passages and figure illustrate a separate and distinct memory that is not perturbed by the profile. If the profile data is stored in the kernel space then the shared user memory is considered the separate memory or at very least the separate and distinct memory is the removable media unit 170 and the permanent storage 160. This is the broadest reasonable interpretation.

Otherwise, **Klapproth** demonstrated that it was known at the time of invention to provide a separate and distinct memory for tracing and profiling (column 2, lines 3-40, figure 1, element 58, column 3, lines 16-21). It would have been obvious to one of ordinary skill in the art at the time of invention to implement trace/profiling system of **Krishnaswamy** with a separate buffer for storing as found in **Klapproth**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide an interface that doesn't unduly burden the bus (column 2, lines 32-33).

Claim 4

Krishnaswamy disclosed the method according to claim 1, wherein said updating step is triggered by execution of the events *(column 6, lines 21-33)*.

Claim 5

Krishnaswamy did not explicitly state the method according to claim 1, wherein said updating step is triggered by execution of instructions embedded into an instruction

stream of the computer program. **Krishnaswamy** demonstrated that it was known at the time of invention to instrument code for profiling (column 1, lines 56-57). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the profiling-based optimizing compiler of **Krishnaswamy** with instrumented code as found in **Krishnaswamy**'s own teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow for collection of a minimum amount of data, thus saving space and time (column 1, lines 60-62), additionally not all processors are equipped with performance monitoring functions and thus instrumentation is required for profiling.

Claim 6

Krishnaswamy disclosed the method according to claim 1, further comprising the step of detecting whether a profile count has exceeded an adjustable predefined threshold *(column 6, lines 30-34)*.

Claim 7

Krishnaswamy disclosed the method according to claim 1, further comprising the step of indicating when a profile count has exceeded an adjustable predefined threshold *(column 6, lines 30-34)*.

Claim 8

Krishnaswamy disclosed the method according to claim 7, wherein said indicating step comprises the step of raising an exception *(column 6, lines 30-34)*.

Claim 11

Krishnaswamy disclosed the method according to claim 1, further comprising the step of identifying profile information corresponding to the profile counts using a profiling event identifier *(column 6, lines 26-36; column 1, lines 34-43; identification of some sort required for proper usage of collected information)*.

Claim 12

Krishnaswamy disclosed the method according to claim 11, further comprising the step of addressing the memory array, using the profiling event identifier *(column 6, lines 24-36; column 1, lines 34-43; identification of some sort required for proper usage of collected information)*.

Claim 13

Krishnaswamy disclosed the method according to claim 1, further comprising the steps of: generating the profile counts using profile counters associated with the events *(column 6, lines 24-28)*. **Krishnaswamy** did not explicitly state maintaining the profile counters in a set-associate manner. Official Notice is taken that it was known at the time of invention to store values in a set-associative manner. It would have been

obvious to one of ordinary skill in the art at the time of invention to implement the memory of **Krishnaswamy** with a set associative manner. This implementation would have been obvious because one of ordinary skill in the art would be motivated to make use of a regular method of memory, which thus common and easy to use/implement.

Claim 16

Krishnaswamy disclosed the method according to claim 1, further comprising the step of supporting read operations from the memory array in an off-line optimization of the program *(column 1, lines 30-43)*.

Claim 17

Krishnaswamy disclosed the method according to claim 1, further comprising the step of assisting optimization of the program, based upon the profile counts stored in the memory array *(column 1, lines 34-37)*.

Claim 22

Krishnaswamy disclosed the method according to claim 1, wherein the memory hierarchy includes data and instruction caches, and the memory array is separate and distinct from the memory hierarchy so as to not perturb normal operations of the data and instruction caches *(Figure 2; and as above for claim 1)*.

Claim 38

Krishnaswamy disclosed the method according to claim 1, wherein said method is implemented by a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform said method steps *(column 1, lines 34-37; compiler)*.

4. Claims 3, 9-10, 23-30, 32-34, 37 and 39 rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of "Dictionary of **Computing**".

Claim 3

Krishnaswamy did not explicitly state the method according to claim 1, wherein said storing and updating steps are performed asynchronously to prevent a decrease of an execution speed of the computer program. **Computing** demonstrated that it was known at the time of invention to perform circuit operations asynchronously (page 26, asynchronous circuit). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the system of **Krishnaswamy** with an asynchronous circuit design, including storing and updating counts as suggested by **Computing**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to allow operation with a minimum of delay (page 26, asynchronous circuit).

Claim 9

Krishnaswamy disclosed the method according to claim 1, further comprising the steps of: accumulating profile updates (**Krishnaswamy**: column 1, lines 34-37). **Krishnaswamy** did not explicitly state dividing the accumulated profile updates by a threshold fraction. **Computing** demonstrated that it was known at the time of invention to make use of scaling (page 432). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the profiling counters of **Krishnaswamy** with scaling (or dividing/multiplying by a threshold fraction) the update value as found in **Computing**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to adjust the stored value to the hardware/equipment (register size) limitations (**Computing**: page 432).

Claim 10

Krishnaswamy did not explicitly state disclosed the method according to claim 1, further comprising the step of scaling the profile counts to prevent profile information overflow. **Computing** demonstrated that it was known at the time of invention to make use of scaling (page 432). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the finite hardware memory of **Krishnaswamy** with scaling the update value as found in **Computing**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to adjust the stored value to the hardware/equipment (register size) limitations (**Computing**: page 432).

Claim 23

The limitations of claim 23 correspond to the limitations of claims 1 and 10 and as such are rejected in the same manner.

Claims 24-30, 32-34, 37 and 39

The limitations of claims 24-30, 32-34, 37 and 39 correspond to the limitations of claims 3-9, 11-13, 22 and 17 and are dependent upon claim 23. Thus, the claims are rejected in the same manner as 3-9, 11-13, 22 and 17 in consideration of claim 23.

5. Claims 14-15 rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Record** et al. (USPN 5,355,484).

Claims 14 and 15

Krishnaswamy did not explicitly state the method according to claim 13, further comprising the step of selecting a profile counter to be evicted from the memory array based upon a predefined replacement, when a number of profiling events assigned to an associative class of events is exceeded. **Record** demonstrated that it was known at the time of invention to perform the above operation (column 9, lines 13-20). **Record** further demonstrated (as found in claim 15) that it was known at the time of invention to utilize the replacement strategy based upon on of least-recently-used and first-in-first-out (column 9, lines 13-20). It would have been obvious to one of ordinary skill in the art

at the time of invention to implement the optimizing profiling system of **Krishnaswamy** with the control provided by **Record**. This implementation would have been obvious because one of ordinary skill in the art would be motivated to minimize any reduction in execution time resulting from profiling a system by limiting the number of events to be monitored (**Record**: column 2, lines 17-25).

6. Claims 35 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of Dictionary of **Computing**" in further view of **Record** et al. (USPN 5,355,484).

Claims 35-36

The limitations of claims 35 and 36 correspond to the limitations of claims 14 and 15 and are indirectly dependent upon claim 23. Thus, the claims are rejected in the same manner as 35 and 36 in consideration of claim 23.

7. Claims 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Altman** et al., "DAISY: Dynamic Compilation for 100% Architectural Compatibility".

Claim 18

Krishnaswamy did not explicitly state the method according to claim 1, wherein said assisting step is performed during at least one of dynamic binary translation and

dynamic optimization [compilation] of the computer program. **Altman** demonstrated that it was known at the time of invention to provide dynamic binary translation and dynamic optimization [compilation] (page 27, section 2 and page 28, section 2.1; additionally page 27, left column, last three paragraphs). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the profiling compiler system of **Krishnaswamy** with dynamic translation and optimization [compilation] as found in **Altman**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide compiling/translating system with dynamic operation (useful for providing real-time operation; page 27, left column, second and third paragraphs) and profiling for optimization (useful for helping code execute better).

Claim 19

Krishnaswamy and **Altman** disclosed the method according to claim 18, wherein the dynamic binary translation and dynamic optimization of the computer program results in translated and optimized code, respectively, the translated and optimized code comprising instructions groups which pass control there between *(**Krishnaswamy**: column 1, lines 30-43; and **Altman**: page 27, right column, third paragraph; page 29, section 3).*

8. Claims 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Altman** et al., "DAISY:

Dynamic Compilation for 100% Architectural Compatibility" in further view of **Chang** et al., "Using Profile Information to Assist Classic Code Optimizations".

Claims 20 and 21

Krishnaswamy and **Altman** did not explicitly state the method according to claim 19, further comprising the step of identifying frequently executed paths of the computer program, by instrumenting exits from the instruction groups with a profiling instruction that indicates a unique group exit identifier. **Chang** demonstrated that it was known at the time of invention to instrument groups of instructions to provide an ID (page 1305-1306, item (a) under "Profiler implementation") and to optimize frequently executed paths (page 1306, bottom). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the optimizing profiling compiler of **Krishnaswamy** and **Altman** with group instrumentation as found in **Chang**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to optimize frequently executed program paths (page 1301, Introduction). **Chang** did not explicitly state to instrument exit points. Official Notice is taken that it was known at the time of invention to instrument exits. Furthermore, **Chang** demonstrated instrumenting the entry point (page 1305-1306, item (a) under "Profiler implementation") or taken more generally simply ensuring instrumentation of the group/function. It would have been obvious to one of ordinary skill in the art at the time of invention to instrument exits of groups/functions in the compiler of **Krishnaswamy**, **Altman** and **Chang**. This implementation would have been obvious because one of

ordinary skill in the art would be motivated to provide an information about the profiled code, which includes determining if a group/function is executed. Both entry and exit points are the most obvious instrumentation points of all locations, since the are easily identifiable. Additionally, **Krishnaswamy** and **Altman** did not explicitly state the method according to claim 19, further comprising the step of extending the instruction groups along a frequently executed path. However, **Chang** demonstrated this as well on page 1306, items (b) through (e) and page 1301-1302, "Introduction".

9. Claims 40 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Chang** et al., "Using Profile Information to Assist Classic Code Optimizations".

Claim 40

Krishnaswamy disclosed a method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy, comprising the steps of:

- executing a computer program *(column 1, lines 33-35)*;
- storing, in a single memory array, a plurality of event-specific profile counts, each associated with an event associated with the execution of a path of the computer program *(column 1, lines 36-39)*

- selecting at least one of a plurality of event-specific profile counts for profiling the path of the computer program *(column 6, lines 24-28, at some point in time the events to be profiled are selected)*
- updating the profile counts for only the events *(column 1, lines 33-37)*

Depending on interpretation of "the memory array being separate and distinct from the memory hierarchy", **Krishnaswamy** may have anticipated the claimed invention through column 5, lines 35-45, column 6, lines 34-36 and figure 2, elements 70, 100, 110, 160 and 170. The passages and figure illustrate a separate and distinct memory that is not perturbed by the profile. If the profile data is stored in the kernel space then the shared user memory is considered the separate memory or at very least the separate and distinct memory is the removable media unit 170 and the permanent storage 160. This is the broadest reasonable interpretation.

Krishnaswamy did not explicitly state if at least one of the selected event-specific counts has exceeded a predefined threshold, optimizing the portions of the computer program associated with the event-specific profile counts more aggressively than other portions of the computer program. **Chang** demonstrated that it was known at the time of invention to optimize more heavily various parts of a program based upon threshold (pages 1306-1308, "Optimizing frequently executed paths" section). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the optimizing profiling compiler of **Krishnaswamy** with group instrumentation as found in

Chang's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to optimize frequently executed program paths primarily since they are executed more (page 1301, Introduction and pages 1306-1308).

Claim 42

Krishnaswamy did not explicitly state the method according to claim 40, wherein the memory array is arranged as a two-way set associative array. Official Notice is taken that it was known at the time of invention to store values in a two way set-associative manner. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the memory of **Krishnaswamy** with a set associative manner. This implementation would have been obvious because one of ordinary skill in the art would be motivated to make use of a regular method of memory, which thus common and easy to use/implement.

10. Claims 41 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Chang** et al., "Using Profile Information to Assist Classic Code Optimizations" in further view of **Altman** et al., "DAISY: Dynamic Compilation for 100% Architectural Compatibility".

Claim 41

Krishnaswamy did not explicitly state the method according to claim 40, further comprising the step of optimizing the computer program during at least one of static and dynamic compilation using the profile information. **Altman** demonstrated that it was known at the time of invention to provide dynamic binary translation and dynamic optimization [compilation] (page 27, section 2 and page 28, section 2.1; additionally page 27, left column, last three paragraphs). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the profiling compiler system of **Krishnaswamy** with dynamic translation and optimization [compilation] as found in **Altman**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to provide compiling/translating system with dynamic operation (useful for providing real-time operation; page 27, left column, second and third paragraphs) and profiling for optimization (useful for helping code execute better).

11. Claims 3, 9-10, 23-30, 32-34, 37 and 39 rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354) in view of "Dictionary of **Computing**". Rejection of above not repeated here.

12. Claims 14-15 rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354) in view of **Record** et al. (USPN 5,355,484). Rejection of above not repeated here.

13. Claims 35 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354) in view of Dictionary of **Computing**" in further view of **Record** et al. (USPN 5,355,484). Rejection of above not repeated here.

14. Claims 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354) in view of **Altman** et al., "DAISY: Dynamic Compilation for 100% Architectural Compatibility". Rejection of above not repeated here.

15. Claims 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354) in view of **Altman** et al., "DAISY: Dynamic Compilation for 100% Architectural Compatibility" in further view of **Chang** et al., "Using Profile Information to Assist Classic Code Optimizations". Rejection of above not repeated here.

16. Claims 40 and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN

5,590,354) in view of **Chang** et al., "Using Profile Information to Assist Classic Code Optimizations". Rejection of above not repeated here.

Claim 40

Krishnaswamy disclosed a method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy, comprising the steps of:

- executing a computer program *(column 1, lines 33-35)*;
- storing, in a single memory array, a plurality of event-specific profile counts, each associated with an event associated with the execution of a path of the computer program *(column 1, lines 36-39)*
- selecting at least one of a plurality of event-specific profile counts for profiling the path of the computer program *(column 6, lines 24-28, at some point in time the events to be profiled are selected)*
- updating the profile counts for only the events *(column 1, lines 33-37)*

Klapproth demonstrated that it was known at the time of invention to provide a separate and distinct memory for tracing and profiling (column 2, lines 3-40, figure 1, element 58, column 3, lines 16-21). It would have been obvious to one of ordinary skill in the art at the time of invention to implement trace/profiling system of **Krishnaswamy** with a separate buffer for storing as found in **Klapproth**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would

be motivated to provide an interface that doesn't unduly burden the bus (column 2, lines 32-33).

Krishnaswamy did not explicitly state if at least one of the selected event-specific counts has exceeded a predefined threshold, optimizing the portions of the computer program associated with the event-specific profile counts more aggressively than other portions of the computer program. **Chang** demonstrated that it was known at the time of invention to optimize more heavily various parts of a program based upon threshold (pages 1306-1308, "Optimizing frequently executed paths" section). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the optimizing profiling compiler of **Krishnaswamy** with group instrumentation as found in **Chang**'s teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to optimize frequently executed program paths primarily since they are executed more (page 1301, Introduction and pages 1306-1308).

17. Claims 41 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Krishnaswamy** et al. (USPN 6,622,300) in view of **Klapproth** et al. (USPN 5,590,354) in view of **Chang** et al., "Using Profile Information to Assist Classic Code Optimizations" in further view of **Altman** et al., "DAISY: Dynamic Compilation for 100% Architectural Compatibility". Rejection of above not repeated here.

Response to Arguments

18. Applicant's arguments filed 14 July 2005 have been fully considered but they are not persuasive. Most of Applicant's arguments are moot in view of the new rejection above. Applicant further argued: 1) no motivation for scaling from Dictionary; and 2) motivation for Chang. The arguments are not persuasive. First, Dictionary clearly states scaling necessary for data to be in range of the equipment. The above motivation state is derived directly from this. Second, merely stating there is no motivation does not prove it, especially in light of a motivation statement. Thus, having addressed all of Applicant's concerns with the above arguments and rejections, the rejections are maintained.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (571)-272-3736. The examiner can normally be reached 9:00am - 5:30pm Monday thru Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)-272-3719. The fax phone numbers for the organization where this application or proceeding is assigned are (571)273-8300 for regular communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703)305-3900.

William H. Wood
October 3, 2005

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100